Package ‘AnDE’

February 19, 2015

Title An extended Bayesian Learning Technique developed by Dr. Geoff Webb

Description AODE achieves highly accurate classification by averaging over all of a small space.

Version 1.0

Author Sai Teja Ranuva & Nayyar Zaidi

Maintainer Sai Teja Ranuva <ranuva.teja@gmail.com>

Depends R (>= 2.14.0), discretization, functional, foreign, stringr

License GPL-3

Collate 'aode.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2013-07-25 12:43:59

R topics documented:

aode ................................................................. 2
calWeight ......................................................... 3
cutX ............................................................... 3
discretizer ......................................................... 4
distributionForInstance ....................................... 5
indexCalc ......................................................... 5
mdl ............................................................... 6
predict.AODE ................................................... 7
setVar ........................................................... 8
training ......................................................... 8

Index 10
Description

This function builds the model using the AODE algorithm which can then be used for classification.

Usage

```r
aode(train, mestimate = 1, weighted = FALSE,
     subsumption = FALSE, S = 100)
```

Arguments

- `train`: data.frame : training data. It should be a data frame. AODE works only with discretized data. It would be better to discretize the data frame before passing it to this function. However, `aode` discretizes the data if not done beforehand. It uses an R package called discretization for the purpose. It uses the well-known MDL discretization technique. (It might fail sometimes)
- `mestimate`: optional numeric
- `weighted`: optional boolean
- `subsumption`: optional boolean
- `S`: optional numeric subsumption constant

Details

This is the training phase of the algorithm. Necessary count and probability tables are generated which will be used for the prediction purpose.

Value

An object of class AODE

Author(s)

saiteja ranuva

Examples

```r
require("datasets")
aode(iris, mestimate=1)
aode(iris)
aode(iris, weighted=TRUE)
```
### calWeight

**Description**

This function calculates mutual information between the attr and class.

**Usage**

```r
calWeight(aode)
```

**Arguments**

- `aode` list. this is the list which has all the required variables in it.

**Details**

This function is called when weighted flag is set

**Value**

- `aode` list. updated value

**Author(s)**

saiteja ranuva

---

### cutX

**Description**

This function takes in a data frame to be discretized. The data type of the columns are important. Only the numeric columns are discretized.

**Usage**

```r
cutX(data, cutp)
```

**Arguments**

- `data` data.frame. This data frame is discretized and returned.
- `cutp` list - A list of cutp points obtained from training data

**Details**

This uses the cut points generated while discretizing training data to discretize test data.
discretizer

Value

data data.frame. This the discretized data frame

Author(s)

saiteja ranuva

Description

This function takes in a data frame to be discretized. The data type of the columns are important. Only the numeric columns are discretized.

Usage

discretizer(data)

Arguments

data data.frame. This data frame is discretized and returned.

Details

Here we use Fayyad’s mdl discretization method. Discretizing data by MDL method as implemented in the package 'discretization'

Value

data data.frame. This the discretized data frame

Author(s)

saiteja ranuva
**distributionForInstance**

*Description*

predicts class of a given instance based on the model

*Usage*

```
distributionForInstance(x, aode)
```

*Arguments*

- **x**: instance to be classified
- **aode**: list. this is the list which has all the required variables in it.

*Details*

details to be added

*Value*

class integer predicted class of the instance

*Author(s)*

sai teja ranuva

---

**indexCalc**

*Description*

Calculates all the array and matrix indices required.

*Usage*

```
indexCalc(aode)
```

*Arguments*

- **aode**: list. this is the list which has all the required variables in it.

*Details*

Details - add later
Description
This function takes in a data frame to be discretized. The data type of the columns are important. Only the numeric columns are discretized.

Usage
mdl(data)

Arguments
data data.frame. This data frame is discretized and returned.

Details
Here we use Fayyad’s mdl discretization method. Discretizing data by MDL method as implemented in the package 'discretization'

Value
list of cut points and the discretized data frame

Author(s)
saiteja ranuva
Description

This is a generic function. This function predicts the class of the test data and returns a vector of predicted values.

Usage

```r
## S3 method for class 'AODE'
predict(object, test, ...)
```

Arguments

- `object` object of class AODE
- `test` test data frame. If the training data was discretized, then the same cut points shall be used to discretize the test data. So obviously if the training was not discretized, test data should also not be discretized.
- `...` extra arguments which might be needed in future

Details

Written in line with the E1071 package.

Value

class vector containing the predicted class distribution of the test data

Author(s)

sai teja ranuva

Examples

```r
data<-iris
doce<-aode(data)
poect(ode,iris)
```
Description
sets the required variables

Usage
setVar(aode)

Arguments
aode list. this is the list which has all the required variables in it.

Details
calculates the required space for indices and allocates

Value
aode list updated value of the list

Author(s)
sai teja ranuva

Description
Calculates the count matrices.

Usage
training(aode)

Arguments
aode list. this is the list which has all the required variables in it.

Details
class parent count and class parent child count are calculated
training

Value
  aode list. updated value

Author(s)
  sai teja ranuva
Index

aode, 2

calWeight, 3
cutX, 3

discretizer, 4
distributionForInstance, 5

indexCalc, 5

mdl, 6

predict.AODE, 7

setVar, 6, 8

training, 8